

UNIVERSIDAD RAFAEL LANDÍVAR
FACULTAD DE INGENIERÍA
INGENIERÍA EN INFORMÁTICA Y SISTEMAS
LENGUAJES FORMALES Y AUTÓMATAS
CATEDRÁTICO: ING. MOISÉS ALONSO

FASE 1:
ANALIZADOR LÉXICO

FERNANDO OLIVA 1251518

GUATEMALA, 4 DE MARZO DE 2018

INTRODUCCIÓN

Como parte del curso LENGUAJES FORMALES Y AUTÓMATAS, se presenta la primera fase del proyecto que corresponde a la implementación de un Analizador Léxico por medio de expresiones regulares que permitan evaluar la validez de una gramática leída desde un archivo de texto.

Para ello, se hace uso de las expresiones regulares y un árbol de expresiones que permite evaluar cada sección, SET, TOKENS Y ACTIONS, del archivo de texto correspondiente analizando y comparando cada una de las líneas de dichas secciones.

Adicionalmente, si la gramática no fue correcta se muestra en pantalla el posible error que esta tenga, así como la línea del archivo en donde el fallo fue detectado.

OBJETIVOS

General

1. Implementar un analizador léxico por medio de expresiones regulares.

Específicos

2. Verificar la validez de una gramática por medio del analizador léxico
3. Identificar los posibles errores en la gramática analizada.

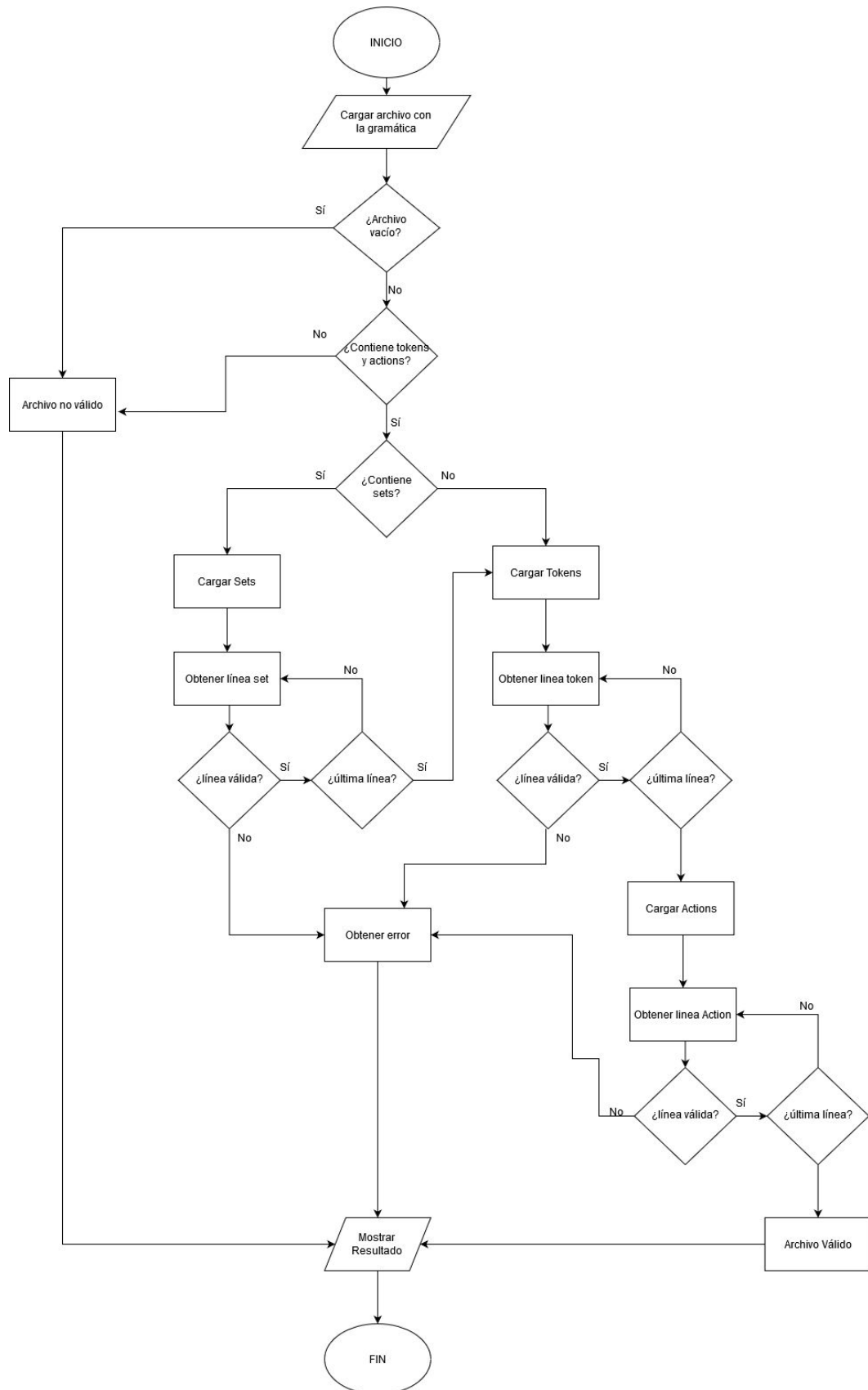
ANÁLISIS

ENTRADAS	<ul style="list-style-type: none">● Archivo de texto con la gramática que se desea evaluar.● Expresiones regulares para cada una de las secciones del archivo (SETS, TOKENS, ACTIONS).
PROCESOS	<ul style="list-style-type: none">● Armar un árbol para cada una de las expresiones regulares.● Recorrer el archivo de texto para obtener cada una de las secciones del mismo.● Evaluar cada sección del archivo con la expresión regular correspondiente.
VALIDACIONES	<ul style="list-style-type: none">● El archivo de texto no debe venir vacío.● La expresión regular no debe ser nula.● Puede existir “n” cantidad de espacios, saltos de línea y tabuladores entre las secciones del archivo o en los elementos de cada sección.● La sección SETS puede o no existir dentro del archivo. Si existe esta debe contener más de un Set.● Cada uno de los elementos de SETS debe empezar con un identificador único (<id>).● Las secciones TOKENS y ACTIONS siempre deben existir en el archivo de texto.● Cada uno de los elementos de TOKENS debe contener la palabra TOKEN y un número que permita identificarlos, estos números debe ser únicos.● La sección ACTIONS debe contener la función RESERVADAS.● La sección ACTIONS puede tener más de función y cada función debe estar dentro de llaves ({,}).● Cada elemento que pertenece a una función debe contener un identificador único, un número, seguido de la definición correspondiente.

	<ul style="list-style-type: none"> • Cada uno de los elementos de cada sección debe cumplir con la expresión regular correspondiente. • La sección ERROR puede o no existir, si existe debe estar después de la definición de la última función de ACTIONS.
SALIDAS	<ul style="list-style-type: none"> • El archivo contiene una gramática correcta o incorrecta. • Si la gramática es incorrecta se muestra la línea del archivo que contiene el error.

DISEÑO DEL PROYECTO

DIAGRAMA DE FLUJO



EXPRESIONES REGULARES

Para que las expresiones sean válidas previamente deben ser retirados los espacios, saltos de línea y tabuladores que lleguen a existir en cada elemento de cada sección. “\w” representa cualquier símbolo.

Para la sección SETS se utiliza la expresión regular:

`[A-Z]+ [=] (([\w[^\.\]\w[^\]] | ([\w[^\]] | ([CHR][0-9]+[.CHR][0-9]+))+ [+]?`

Para la sección ACTIONS se utiliza la expresión regular:

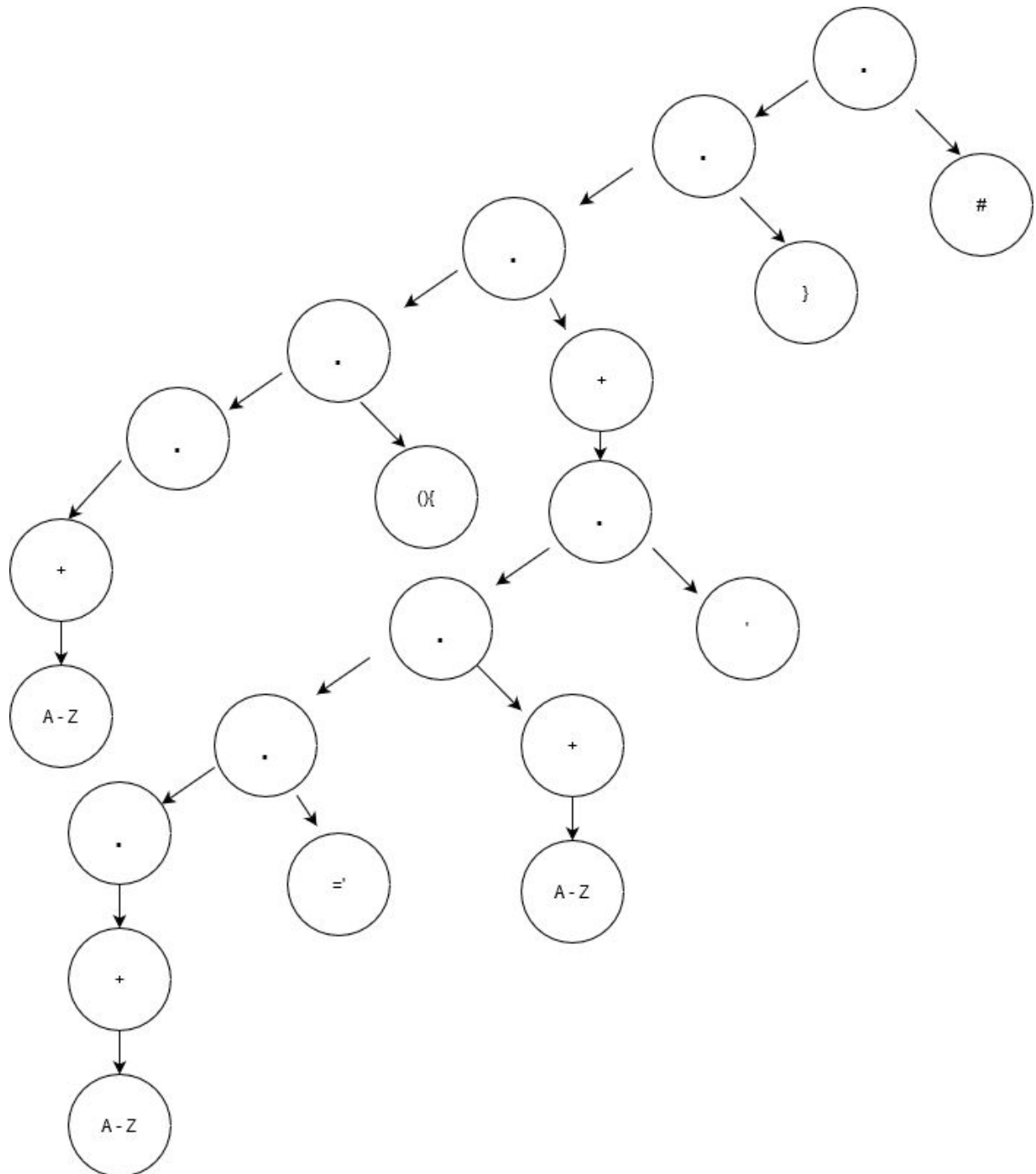
`[A-Z]+ [() { } ([0-9]+ [='] [A-Z]+ ['])+ [}]`

Para la sección TOKENS se utiliza la expresión regular (previamente deben haber SETS Y ACTIONS):

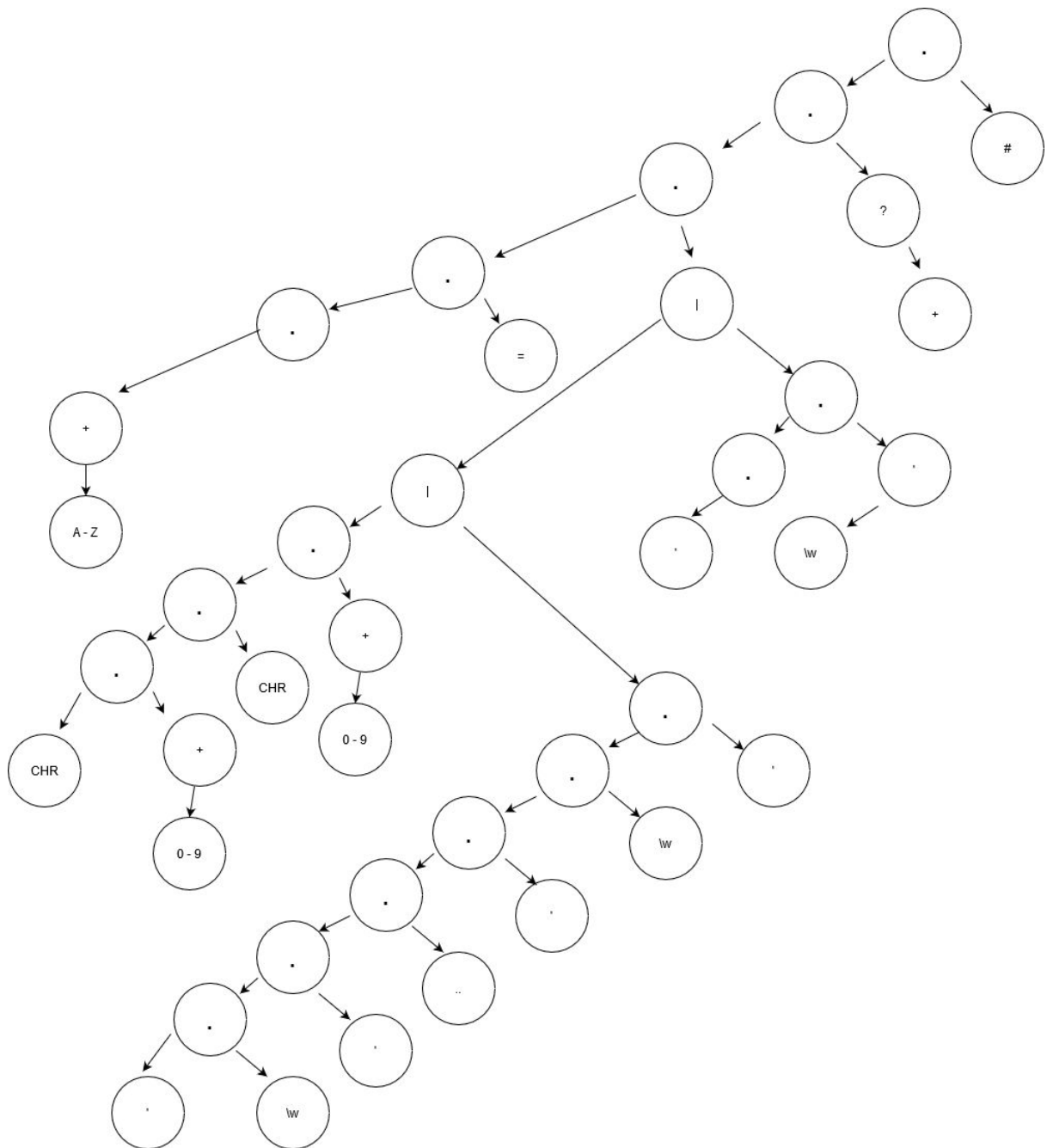
`[TOKEN] [0-9]+ [=] (([\w[^\]]+ | ((SET | ACTION)* ([(] | [)]) | [{] [}] | [+] | [*] | [|] | [?])*)+`

ACTIONS:

ACTIONS:



SETS



TOKENS

